LOADSTAR LETTER #67

Two Heads are Better Than One

By John Elliott. Note: What follows assumes you are using a disk with only one notch. "Flippies" (two notches), are designed to be flipped over so that each side can be used separately.

If the instructions on page 87 of the 1571 disk drive manual are followed, the user will likely destroy the drive. When I tried the recommended command, it did not work. However, it sounded as if an angry woodpecker was trapped inside. I will not copy here the incorrect command.

The 1571 has a top and bottom head. Unlike the 1541, it can read both sides of a disk without turning it over. When connected to a 128, the first computer command sets the drive automatically to two headed or 1571 mode. This means that both sides of the disk form one continuous directory. The "back" side of the disk formatted in 1571 mode cannot be read by a 1541 drive. It is also invisible to a 1571 attached to a c64 when the disk remains in the default single sided or 1541 mode.

There are however special drive commands that can place a 1571 in single or double-sided mode with either a c64 or 128.

All commands should be preceded with "open1,8,15". Commands should be placed in quotes, and followed by ": close15". For instance open1,8,15,"u0>m0":close15. If you have an application such as a word processor that accepts drive commands at a prompt, enter just the command that I have in quotes, but do not use the quotation marks.

The above command sets the drive for single sided (1541) mode.

"u0>m1" sets for double-sided (1571) mode.

"u0>h0" selects the "front" side of the disk (head zero).

"u0>h1" selects the "back" side of the disk (head one).

Note: You can only switch sides when in single sided mode (u0>m0). The manual is wrong on this point.

Formatting on a 128: If you wish to format the 1571 disk as one continuous directory (1571 mode), and the drive is connected to a 128, simply use the standard "n:disk name,01" approach. If you want to separately format both sides of the disk, without turning it over, and are using the 128, you must first issue the "u0>m0" command. You will default to, and can format, head zero, "front" side. To use the back, send a "u0>h1" command. Then format.

Formatting on a 64: If the drive is connected to a c64, the default is single sided. It is already in "u0>m0" mode. Format as described above. To format as one continuous directory (1571 mode), send the "u0>m1" command. Then format with the standard command.

A Formatting Problem: If a disk is already formatted as double sided, you can reformat the front after first resetting as single sided (u0>m0). You will not be able to reformat the backside (head one).

There are at least four solutions to this problem:

1) Reformat the entire disk as ms dos with Big Blue Reader or Little Red Reader. Then format side zero as normal. Change heads and format side one.

2) After formatting side zero, remove the disk; send the head change command (u0>h1); reinsert the disk and format side one.

3) Since the problem is that the "back" side is being initialized, avoid initialization with the disk left in,

a) using instead of ">" the "-" (hold the Commodore key and press v), or

b) using "u0"+chr$(128+62) +"h1" Formatting in 1571 or 1541 mode: observations

Single sided or 1541 mode formatting is an audibly slower process than a 1571 format. JiffyDOS detects whether the disk is formatted as 1541 or 1571 and acts accordingly. Unless a "flippy" is turned over, its "back" cannot be read by a 1571 in either mode even if (u0>h1) is used.

Advantages of either mode: If in m0 mode, the total storage

© 1999 by J & F Publishing, Inc. The LOADSTAR LETTER is published monthly by J&F Publishing. 606 Common Street, Shreveport LA 71101. Subscription rate is $18.00 in the USA 12 issues. $20 outside of the USA. No part of this newsletter may be reproduced without the permission of J & F Publishing. Contacts:
jeff@LOADSTAR.com
US MAIL: ATTN. Jeff Jones
J & F Publishing
P.O. Box 30008 Shreveport
LA 71130-0008
Phone: 318/221-8718,
Fax: 318/221-8870

space between the two sides is 644 blocks times two, or 1288 blocks. If the m1 mode is used the continuous directory's capacity is 1328 blocks, or 40 extra blocks.

Since m1 gives only one directory, a maximum of 144 files can be stored on the disk. The m0 command provides between the two sides 144 times 2 or 288 files. I recently had to stop using a disk when it was only half full, because I had saved a very large number of quite small graphics files for my 128. If I had separately formatted the two sides, I would still be using that disk.

If in m0 mode you separately format the two sides of a disk, you can have a secure directory that is not immediately visible. This increases privacy. If you accidentally reformat one side of the disk in m0 mode, the other side will survive. You can save two copies of the same file to the same disk without changing names by switching heads.

Helpful Software: I have very few programs or cartridges that use the two-headed possibilities of the 1571.

Warp Speed has a command that will switch between single and double-sided mode. Andrew Vardy has uploaded to ftp.funet. fi, "disk copier 64". It will automatically copy one side of a disk to the other side, either within the disk, or between disks. I do not know of a program that will automatically copy individual files from one side to another.

Summary:

1) The 128 defaults to 2 headed (m1), mode. The 64 defaults to single headed (m0) mode.

2) Only if in single headed mode (m0), can the head switch (u0>h0) or (u0>h1) be used to force the drive to read either side of the disk. The manual is wrong on this point.

3) Initialization must be avoided for the "back" side if a m1 mode disk is to be reformatted as m0.

My problems with page 87 of the manual were helped via Internet in roughly equal amounts by Andrew Vardy, Dick Cunningham and Nicolas Welte. My thanks.

Color JPEG Printouts With a Commodore

By John Elliott. Maurice Randall has developed a program that prints jpegs to paper. Depending on the printer, the results will be monochrome or in color. What is seen on the Commodore screen is a rectangle that represents the size and location of the object. The rectangle can be moved and resized.

He and Dale Sidebottom obtain the jpegs from their Sony Mavica 700 digital still cameras. These have a built in 1.4 meg drive. They read the disk with a CMD FD-2000 drive using Little Red Reader on a 128. The Big Blue Reader version 4.1 will also read these higher density disks if the appropriate CMD drive is used. These cameras will not take a double density disk that would be readable by a 1581. He uses a Panasonic KX-P5400 LED monochrome printer. Dale is using a color inkjet to print color impressions of the jpeg.

What the two printers have in common is that they have PostScript Level 2 language built in. Maurice has been able to get the printer rather than the Commodore to do the image translation. His LED printer converts the jpeg colors to half tones to show a very nice grayscale image. Dale's color inkjet gives a full 24bit image from a 16 million-color palet. My Epson Stylus IIs color inkjet printer will not work with this program. It does not have PostScript built in. Nor does it have the 16 megs of ram that Dale's printer has.

This program runs on a 128. I am not sure if there is a c64 version. When it is finished, it will be available from Dale.

Fear And Loathing In Cyberspace

By Jeff Jones. The other day my wife called me at work with a distressing message. Someone had taken $250 out of our account. It was after business hours so we couldn't call the bank. My wife is very adept at using the automated teller system over the telephone and dialed the bank. After pressing a slew of buttons and listening to the voice menus, my wife found out that the $250 deduction in our account was some sort of "fee."

Since we had recently had a six-dollar check to Circle K clear at our bank for over $1000, we were instantly angry. We couldn't wait to snap at a service representative the next morning at one second after 8:00. On the way home I passed our bank and noticed a billboard near it that listed a web site where we could do online banking. I never really got the hang of what my wife does on the telephone. There are simply too many codes to remember, and the voice system doesn't give all information. Like for instance it doesn't list all ATM withdrawals. It also takes a bunch of keystrokes to simply find out which checks cleared. With a real alphanumeric keyboard on the net, I figured I'd be able to navigate my account better.

So I logged on to the net and checked out the web site. I entered my account number and it asked for my password. I had none so my wife, smart thing that she is, told me that it probably wanted my PIN (Personal Identification Number) number. I entered my PIN number and it got me through, but the web site informed me that I had to change my password to something else, and wouldn't let me proceed until I did so.

My account scrolled onto the screen. Every check that had cleared was there for me to see in seconds. But right there, plain to see: A check written to me for $250 as a down payment for wedding photography, had bounced. The name of the bride had been noted for me, and the bank of origin. The "fee" was the bank snatching its $250 away from my account, as they rightfully should do. I told my wife that it was apparently no mistake, which would save us some embarrassment if we called about the account. My wife came upstairs and stared at the screen, looking at the account balance, the cleared checks, the options to transfer money, etc. She frowned and said, "I don't like that!"

"You don't like what?"

"I don't like our account being on the Internet!"

I made a face, "Why?"

"Because anyone could get to this web site and see our account."

I made another face, "No they couldn't. They'd need our account number as well as a long password supplied by me – a word that doesn't appear in the dictionary."

"But still, anyone could get to it."

I sighed, "Rhonda, anyone could get on the phone and do what you do. They could dial the bank, use the *Access24* service and transfer our money – but that's only if they know our account number and access code."

"That's different."

"How?" I challenged.

She paused for a long while. "I just don't like it."

My wife still doesn't like it. I remember not liking it when *Access24* was new to me, but now I realize that no one is going to steal our money online. It's my firm belief that Internet commerce would be booming even more than it is now if the notion of Internet theft weren't so prevalent. Even Microsoft gets in on the paranoia. Their Internet Explorer will warn you every time you answer a question on a form until you disable the paranoia. Sure it's possible that someone could be somehow listening when I buy a book with my credit card from amazon.com, but frankly that's not going to be anyone in Shreveport. That would be a high tech crime, and frankly, I don't know how anyone would pull it off. Of course, anyone with a telescopic microphone could eavesdrop on someone calling out his or her credit card number on a more traditional purchase. Plenty of times I've heard Fender very clearly call out his credit card number from his office. I've also heard others do it. Were I a thief, I might have scribbled down the number and brokered it off. Of course, if Fender had made those purchases online, he would have been in less danger of having his stolen.

My tree in this forest of an editorial is that crime happens. Most criminals are stupid. They have to rob you or dig through your garbage to get to your credit card numbers. The people who could steal your credit card number when you make a purchase online are not stupid – and most likely have a job – and are most likely stealing larger sums of money from companies, not individuals.

Internet Porn and Spam Solutions

By Jeff Jones. I hope the President is reading this article. I train GM and other workers in Microsoft Office applications. Several of my students have complained that there's too much pornographic email on the net, apparently especially on AOL. Also when they get online to do homework and try to search for a subject, seven out of ten returns of their search is porn. When you have a modern PC, you actually see the porn teasers up front. They ask me how to block this stuff from their computers. I immediately tell them that they can't – not if they are going to use their computers for research.

Services like Yahoo will

actually advertise porn sites – but only if they *think* you just searched for porn. If you use a service like Net Nanny in order to protect your kids from sex, you end up blocking all reference to what Net Nanny thinks is sexual. There have been many complaints from women's organizations, especially breast cancer and research, that these child guard services block them out simply because they use obscene terms like "women" and "breast." You can't search for something female without finding a porn site. I have a newscaster friend named Kysa. I thought it would be nice to find a radio or television station with the call letters KYSA. When I searched, all I found was porn stars named Kysa, including an admittedly weird woman at www.kysaonline.com who is deaf, and uses "deaf retrospection" as an excuse for being so weird and sexual.

My suggestion is not so new. I think that porn peddlers have the right to peddle porn, but they need their own district. They need a *new global top level domain.* When I say global top level domain, I mean .com and .net and .org. I think that porn peddlers need to have addresses that end in .sex or .cum or .xxx but I think it should be *law.* www.kysaonline.com should be forced to become www.kysaonline.sex because of its content – even if it is biographical. So when a little boy with raging hormones searches for Mary Curie, he won't find a porn star named Mary. It would be easy to block out all sites that end in .sex. By that same token, people who are out *searching* for sex will find it easier to get to it. Some people say that this violates first amendment rights, but I disagree.

On the subject of spam, there is a simple solution. The reason I get twenty to thirty pieces of Email per day is because my address appears in a list that is sold to email marketers. That's fine. I love junk mail. What I don't like is to be insulted. Don't email me and promise me that if I take your pill, I'll attract women or that if I send you $49.95 you'll send me the secret to wealth. Therefore, I think there should be a major change in how we handle Email in the world.

Right now Email is free. It shouldn't be. I could send one piece of mail or I can send 30 million pieces of mail and still spend nothing. If Email cost even a half-cent per mail, most people could afford it. Alternatively, maybe we could get 100 free Emails per month with unlimited reception, but after our 100th outgoing piece, we should pay a penny per note. Too many people realize that a sucker is born every minute. If they send out 90 million ads for snake oil, they pretty much know that 1% of that 90 million are gullible enough to send money. No one would send mail to 90 million people if they had to pay – not without some market research first. I say that we need to tax Email.

Internet Spam Awards!

Every once in a while I get a piece of spam that's good enough to keep because the angle is so humorous or clever. Again, a

sucker is indeed born every minute. The person who sent me this Email probably purchased a nice car with the money from suckers who believed him. This guy is going to help you fight spam. He commiserates. Maybe he's legit. Maybe what he sells might help – but doesn't he *have* to know that his letter was itself spam? And if he really is a published author, I think he'd catch the error in the third sentence:

Dear Internet User:

To start let me say this, I'm tired of the porn email in my email box and all that click here crap. I'm a published author in the United States, Germany and Japan. I've spend seven hours or more a day for the last two months researching the topic of Spam on the Internet. I've learned a lot about SMTP blocking, reporting Spam, and blocking your email account from being spammed. In this time I have practiced everything I have learned in blocking this crap mail they call

professional advertising and in the last few weeks I have not yet received one piece of Spam. I wrote a full report on this process I have developed and did some research on how to get this information around to the right people. I've contacted many spammers and did some price research, what better way to find the Spam haters then to find the spammer that is spamming them. I've talked to many well known bulkers as they like to be called and we made the agreement that I will get to advertise to there list in return for a profit break. I'm forced to sell my report for the reason to pay the advertising fee and keep me alive but on the bright side you will learn to fight against the horrible porn mail in your email box. Please join me in fighting this horrible advertising method that has been on the Internet for years. Remember, "Together we stand, divided we fall."
To order my detailed report and provide your help in fighting spam send $15 to the address below and lets all, together, put a stop to spam once and for all.

K.C. Smith

Thank you for your support in this matter.

Your friend and best man in fight against spam.

Knees Calhoon Rocks!

By Robin Harbron. Shortly after I met Fender in Chicago last year, he sent me a CD of some music he had recently recorded. I

think he expressed some surprise when he learned that I was a bit of a musician myself – I play bass in a couple of bands with friends here in Thunder Bay, Ontario, and also regularly lead music at my church with my guitar. Most musicians are the same – very eager to share their music with other like-minded people – eager to find out what they think. I thought I'd share some comments on Fender's album.

The "pre-production" CD I received had three different titles on it – Knees' Rock 'N' Roll, Knees Calhoon: Rock and Roll, and Knees Calhoon's First Rock & Roll CD. In case you didn't pick it up, this is an album of original rock music by our very own Fender – a.k.a. Knees Calhoon.

You're not going to hear 90's rock here – in fact, only three of the songs on the album are even as recent as the 80's. The other ten songs were written between 1966 and 1974. I'm not trying to make anyone feel old, but my wife wasn't even born when most of these songs were written! The point here is that these songs are Rock & Roll in the classic sense.

There is a wide variety of musical influence apparent on the album - a bit of Beach Boys, Neil Young, Dire Straits, and a whole lot of blues. I found it all quite accessible – enough variety from song to song that boredom doesn't set in while listening, but enough continuity that you know that you're listening to the same person the whole way through.

The topic of the songs varies dramatically. Some songs are pure fluff, with no real substance at all – "The Girls Back Home" probably being the worst offender. However, this is Rock & Roll – what do you expect? Well, Fender has many witty and rude lines in "Harlem Nocturnal Emission", and explores some conflicting emotions in "Over the Edge." "Repetition" is worth listening to, over and over again. If you need something oxymoronic, "Mean Standard Deviate" will fit the bill.

I always enjoy obscure story telling songs with melancholy, repetitious tunes (I'm not being sarcastic here – I'm serious!) and Fender has two great ones on this album – "Sailor of the Century" and "Death on the Horizon". In fact, the latter song is definitely my favorite on the whole album. Fender figured I'd like the song, as it has a strong Neil Young influence, and Neil is one of my

favorite musicians. Incidentally, Neil Young lived here in my town for a while, and the university I got my degree at (Lakehead University) gave him an honorary degree a few years ago. While we're talking about music and Thunder Bay – did you know that both Paul Schaffer of David Letterman fame and Johnny Green (the fellow who wrote the Batman theme song for the 60's TV show) are from here? Now you know.

The mix (the relative volume levels of the different components in music) that Fender chose didn't set well with me at first, but has grown on me. The drums and bass take a back seat in most songs, while I'm used to the bass being much more prominent in the music I play and listen to. Guitars and vocals are being featured here – and as I listen to the album more, that's fine with me.

Most songs have at least two guitar tracks going on at once. I particularly like the interplay between the two guitars in "Hi (gh) Hello Mary" and the huge organ like sound Fender has on "Death on the Horizon". "Harlem Nocturnal Emissions" has some great fret-tapping work. "My Baby's Gonna Leave Me Yesterday" has some excellent blues playing. Actually, all the songs have very strong guitar work.

Fender has quite a variety of vocal styles as well – a good range, and he is able to make his voice sound the part for the different musical styles. Background (backup) vocals are tastefully and sparingly done.

In the better-safe-than-sorry category: the lyrics in some of the songs contain words that some might find offensive. However, it's nothing you won't hear on the radio – and most of the songs that contain "naughty" words are a bit tongue-in-cheek.

The bottom line: a fun, somewhat diverse, homegrown collection, spanning twenty years of song writing from the one and only Knees Calhoon.

Reader Mail

Dear Mr. Jones,

Hello! Greetings from Iowa, The Beautiful Land Between two Rivers-- the Mighty Mississippi & Muddy Missouri. Thank-You for the Article on VICE, the Commodore C64, VIC-20 and PET Emulator for PC IBM Compatible Computers. Since I can only get on the Internet at the Public Library, I Downloaded version 0142, because it could fit on a 3.5" floppy. Someday I will have to Download version 0160 at My Friends house. It was Great to see the VIC-20 screen again. The VIC-20 was the first Computer I ever had. I still have My VIC-20. I still can program it. The VIC made Computing Fun. I got out my old MAPPING THE VIC book. I made a Simple Program to see if I could program yet. I could.

I would like to hear from other VIC-20 owners. Goodbye for now. Have a VIC-20 multi-colored Happy Day!

Signed: Bulwinkle, A.K.A. Jeff Puffer
E-Mail: jeffpuffer@hotmail.com

Jeff: Glad to hear you appreciated it.

Gleaned from Comp.Sys.CBM: C Cross compiler

Note from Jeff: I came across this article and it caught my eye because I was considering two things:

1. Learning C on my PC
2. Writing more CBM programs.

I'm shocked at myself for not finding about C cross compilers before now. I already *use* a cross assembler. Hopefully you'll feel as motivated as me when you read of the compiler.

David Murray <dwmurray@flash.net> wrote:

I guess I'm living in the old times since I try to write programs directly on the machines. I was just thinking how convenient it would be to write programs on my PC and compile them in a snap only to be run seconds later in my emulator. Is this the way most people write Commodore applications these days? If so, which is the best cross-compiler to use?

I would assume that many people still write programs directly on the machines. Personally, I prefer cross development using cc65/ca65 (which is no wonder, since I'm the author/maintainer:-)

Here are my reasons why I prefer cross development. For some reasoning about the cc65/ca65 compiler/assembler combo see below.

[Part 1]

1. Editor

When cross developing I have my own editor which I use for everything, from writing news articles (like this one) to TeX documents and C++ code. So I do not have to learn/switch editors, which means I'm productive from the first moment on. My editor has an extension language and I have written a small ca65 mode module which means that I have syntax highlighting when writing assembler code. Not that this is really essential, but it's nice to have.

2. Other tools
In real life I'm an engineer, however I'm earning most of my money by software development. This means I rely heavily on the usual programmers tools: make, grep, awk, man pages and other online docs, access to the web and more. When cross developing all these tools are readily available. I can browse the fridge for a piece of code and paste it straight into the editor. I can use perl to convert binaries into assembler source. I can use make to manage large projects.

3. Speed
While most people are talking about speed when developing on the target machines itself, speed is of no concern when cross developing. The complete Elite 128 code translates in a few seconds on an old P166 running Linux (current size of the resulting binaries is ~70K not counting different language modules, so it is probably one of the larger projects out there).

4. Debugging
Using an emulator, debugging is much easier. I'm using a beta version of VICE with a small patch applied that enables the use of labels that start with an underscore (I'm not sure if this change is in the current version, one of the developers said it will be, but I looked at several betas that don't had it). For debugging, just load the program into VICE, load the symbol table and start debugging. The VICE monitor has some minor problems, but it is much more capable than any (software-) debugger on the real machine. For example, you may debug interrupt handlers or set watch points on variables (break if the variable is read or written to). Using watch points allowed me to find several memory corruption bugs in my programs without any hassle.

5. Ease of use
On the PC I have command line completion, command line recall, shell scripts (named batch files on older systems) and more. Sometimes my PC says "You have mail" and sometimes I'm starting sidplayer to listen to a SID tune in the background.

Most people view the cc65/ca65 combo as a C compiler. While this is not wrong, it is not the full truth. In fact, ca65 is one of the most advanced cross assemblers available. Neither of the advanced features is really essential. You may write even large programs without macros or relocative object modules, but having these advanced features makes life much easier in some situations - and they don't harm you if you don't use them.

Here is a list of the more advanced features:

* Conditional assembly (does this count as "advanced"?).

* Parametrized macros.

* Nested lexical levels and cheap local symbols.

* Creates relocative object modules that are linked together.

* Archiver to handle libraries.

* Segmented architecture. Supports up to 254 segments per object file and up to 65534 segments per application. You may have multiple code-, data- bss-, zero page-, whatever- segments.

* Extremely flexible linker. The linker may be controlled by a configuration file and each segment may be assigned to a separate memory area and/or output file. Together with the segment feature, this allows things like ROMable code, output for splitted ROM areas, overlays (relocate several modules to the same start address) and more. Segments may be filled, aligned, overlaid, they may have different run and load addresses and much more. You may generate several output files in one linker run. For example, the Elite 128 boot code needs to access labels inside the Elite code but apart from that it's a completely separate module and goes into a separate output file. There are default configurations built into the linker, so you don't have to mess with the config files if you don't need them.

* The assembler stores meta information into the object files that allows the linker to emit verbose error messages. Instead of undefined external: _foo

you get

undefined external `_foo' referenced in:
irq.s(61)
lowmem.s(124)

* Full expression support in the linker(!). Try this one with your favorite assembler:

```
.importzp sym1, sym2

lda #((sym1 - sym2)/2)*3
```

This expression cannot be resolved at assembly time, so ld65 will resolve it at link time. And the linker supports *any* expression, that is allowed by the assembler, with any number of external or internal symbols. The linker will even give you verbose error messages like

Error: Range error in module `test1.s', line 3

if the expression above does not fit into a byte (see the topic about meta information above).
[If you want to know, how this is done: The assembler stores the complete expression tree in the object file.]

* Support for the 65SC02 and 65SC816 CPUs. There is not only support in the assembler for the additional instructions but also in the linker. You may relocate 65816 code (or just some segments) to addresses >= $10000.

* Assembler code integrates nicely with code generated by the C compiler.

* The sources are portable: There are binaries available for Unixes, DOS, OS/2 and Windows 95/NT.

* And last, but not least: The code for all tools (assembler, archiver, linker) is freeware and *not* GPLed, so you may use it without restriction in your own projects.

So what about the C compiler (cc65)?

It seems that some people have had expectations that a C compiler for the 6502 cannot fulfill. So here's what you can *not* expect from a 6502 C compiler or from cc65, and why:

* The C standard limits some of the designer's choices. There are several things, a standard conform C compiler has to implement that will give bad code on a 6502 architecture. Some examples are:

- Ints must have at least 16 bits, longs must have at least 32.

- Integer propagation: In an expression, char values must be propagated to integers before expression evaluation. Worse: The 6502 does not support sign extension, so propagating signed chars is rather costly.

- Boolean values are represented by ints.

- Local variables go onto a stack (this is not mandatory according to the ISO standard, but there's no other simple way to make functions reentrant). The stack must be emulated in software on the 6502.

* While C functions are reentrant (may be called by itself), they may not be called from inside an interrupt handler that interrupted C code. This means, that interrupt handlers may not be written in C, at least not without some assembler support.

* cc65 was originally written to run as a cross compiler *and* native compiler. Because of that, it features a very simple code generation strategy. There is no intermediate code and the symbol table is flat (that means there are just two lexical levels). The code generated by the current compiler is several times faster than it was in the original version. However, the code for many constructs is still bad and the current design is max'ed out, so there will be no improvements without a major rewrite.

* While in some areas the library support is pretty complete, several things are still missing. Examples are file routines and support for floating point data types.

That said, it is clear that writing system software or other low level stuff using cc65 is probably a bad idea. On other platforms you may do without assembler if you use C, for the 6502 this is definitely not true (regardless of the C compiler you use). But nevertheless there are lots of projects that may be done using C. Don't let me deceive you by the list above! There are quite a few arguments in favor of writing stuff in C:

* The generated code, while being slow compared to pure, handcoded assembler is actually very fast compared to BASIC. When writing the Morse trainer software (which was written completely in C without a thought about speed or code size), I had problems to get even a simple keying routine running in BASIC because it was too slow, but speed was never a problem when using C.

* C is much more readable than (CBM-) BASIC. You don't have things like line numbers and may add comments everywhere without slowing down your program. Instead of

```
POKE 54308,15 (now, what does this do?)
```

use

```
#define FULL_VOLUME 0x0F
SID.amp = FULL_VOLUME; /* Obvious, isn't it? */
```

And, while the former needs many milliseconds to execute, the later is translated to:

```
lda #$0F
sta $D418
```

* Writing stuff in C is lots faster than writing stuff in assembly. Yes, if you need to take full advantage of the machine, you need assembler, but there are lots of programs that don't need the additional speed. And, writing stuff in C needs about 10-20% of the time than writing the same stuff in assembler (these are my own numbers, your mileage may vary). Many projects were started and never finished because the authors didn't have the time for completion. Again: The first version of the Morse trainer was written in a few hours. Using pure assembler, it would have been days.

* The compiler-generated code integrates nicely with the ca65 assembler. If you reach the limits of C code, just write some routines in assembler. You may design and debug your routines in C, and rewrite them in assembler. To cite the Morse trainer example once more: When adding the feature of parallel keying in the new version, it was clear that this would become messy in C. So I added a small interrupt routine in assembler. The main logic is still in C with all advantages.
This is why it is important to view the cc65/ca65 combination as such, and not only as a C compiler. If C is not enough, you still have one of the most advanced assemblers around.

* Since cc65 is pretty ISO compliant, you may even reuse code written for other platforms. This may give non-optimal code in many places, but it's nice to have this option.

* The code is quite portable between the CBM architectures. Many of my modules compile out of the box for more than one target, or may be easily written to support several targets. This is because many differences are hidden in the runtime library. Just do

```
if (read_joy (JOY_1) &
JOY_FIRE) {
....
}
```

and don't care about the platform specific details. This will also speed up your development (maybe I mentioned that already :-).

To summarize: In my eyes there are many advantages of cross development, and there are advantages writing code in C, especially since you don't loose anything if you have a nice assembler up your sleeves. While the arguments above are obviously biased, you may find one or two valid points. Judge yourself ;-)

If you want to give cc65 a try, I would recommend to use the current snapshot instead of version 2.0 (you will need to compile it yourself, however). The generated code is much better, the library is more complete, and I've fixed several bugs. I'm currently too laz -- busy to release a new version, but the current snapshot is very close. You will find more information at

http://www.von-bassewitz.de/uz/cc65/

Regards

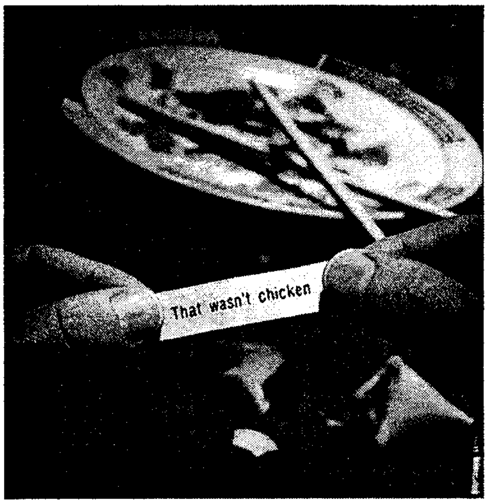
Uz

Ullrich von Bassewitz
uz@musoftware.de

Learn Chinese in 5 Minutes

English phrase Chinese Interpretation

- Are you harboring a fugitive? Hu Yu Hai Ding?
- See me A.S.A.P. Kum Hia Nao
- Stupid Man Dum Gai
- Small Horse Tai Ni Po Ni
- Your price is too high!! No Bai Dam Ding!!
- Did you go to the beach? Wai Yu So Tan?
- I bumped into a coffee table Ai Bang Mai Ni
- I think you need a facelift Chin Tu Fat
- It's very dark in here Wai So Dim?
- Has your flight been delayed? Hao Long Wei Ting?
- That was an unauthorized execution Lin Ching
- Thought you were on a diet Wai Yu Mun Ching?
- This is a tow away zone. No Pah King
- Do you know the lyrics to the Macarena? Wai Yu Sing Dum Song?
- You are not very bright Yu So Dum
- I got this for free Ai No Pei
- I am not guilty Wai Hang Mi?
- Please, stay a while longer. Wai Go Nao?
- Our meeting was scheduled for next week Wai Yu Kum Nao
- They have arrived Hia Dei Kum
- Stay out of sight Lei Lo
- He's cleaning his automobile Wa Shing Ka
- Your body odor is offensive Yu stin ki pu
- Pew! does this bathroom stink! Hu Flung Dung?



What "Win98" Really Means

10. The number of floppies it will ship on.
9. The percentage of people who will have to upgrade their hardware.
8. The number of megabytes you'll have left on your hard drive after you complete the installation.
7. The number of pages in the easy installation summary.
6. The percentage of existing programs that won't run in the new operating system.
5. The number of minutes to install.
4. The number of calls to tech support before you can get it to run.
3. The number of people who will actually PAY for the upgrade.
2. The number of fatal bugs that remain on the release date.

...and the number one thing most people think the *98* in WIN98 stands for:

1. The year it was DUE to ship

LOADSTAR LETTER #67

J&F PUBLISHING 606 COMMON STREET SHREVEPORT LA 71101

- COMMODORE NEWS
- COMMODORE VIEWS